

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ELEKTRONICKÝ ŘÍDICÍ SYSTÉM FOUKACÍ HARMONIKY

ELECTRONIC CONTROL SYSTEM OF MOUTH-ORGAN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Pokorný

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2021

Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Tomáš Pokorný

ID: 211169

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Elektronický řídicí systém foukací harmoniky

POKYNY PRO VYPRACOVÁNÍ:

Komplexně přepracujte elektronický systém pro řízení klávesové foukací harmoniky. Zaměřte se na vestavný systém řízení (Raspberry PI atd.). Systém vybavte nezbytnými akčními členy pro ovládání klapky hudebního nástroje a pro řízení a ovládání tlakového vzduchu. Systém navrhnete pro přehrávání MIDI souborů včetně implementace MIDI kláves.

1. Proveďte literární rešerši a internetový průzkum.
2. Navrhnete a zdůvodnete koncepci systému, definujete zvolené komponenty, navrhnete obvodové schéma.
3. Navrhnete a realizujete funkční vzorek zařízení.
4. Vytvořte programové vybavení pro nadřazený systém i pro vlastní zařízení.
5. Demonstrujte a ověřte plnou funkčnost zařízení a zhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Dle pokynů vedoucího práce.

Termín zadání: 8.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

doc. Ing. Václav Jirsík, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Práce je dokumentace k automatickému elektronicky řízenému systému, který ovládá klávesovou foukací harmoniku. Tato práce popisuje návrh a realizaci výsledného systému

Práce je rozdělena na čtyři části zabývající se postupně teoretickým rozбором, hardwarovým a softwarovým řešením a nakonec realizací.

Klíčová slova

Foukací harmonika, automatický systém, Raspberry Pi, ATmega 165A, Python3, Flask

Abstract

This paper is a documentation for automatic electronic system, that controls mouthorgan with keys. This paper describes design and realization of complete system.

This paper consists of four parts regarding theoretical analysis, hardware, software and finally realization.

Keywords

Mouthorgan, automatic, Raspberry Pi, ATmega 165A, Python3, Flask

Bibliografická citace

POKORNÝ, Tomáš. Elektronický řídicí systém foukací harmoniky. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/134823>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Zdeněk Bradáč.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: Tomáš Pokorný

VUT ID studenta: xpokor77

Typ práce: Bakalářská práce

Akademický rok: 2020/21

Téma závěrečné práce: Elektronický řídicí systém foukací harmoniky

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 23. května 2021

podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce doc. Ing. Zdeňku Bradáčovi Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: 23. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
ÚVOD	10
1. TEORETICKÝ ROZBOR.....	11
1.1 POČÁTEČNÍ STAV.....	11
1.1.1 Foukací harmonika	11
1.1.2 Elektromagnetická táhla Saia-Burgess 195224-223.....	11
1.1.3 Zdroj 24 V.....	11
1.1.4 Regulátor tlaku.....	12
1.2 NÁVRH DPS	12
1.2.1 Schéma zapojení a deska plošného spoje.....	12
1.2.2 Drivery ULN2803	14
1.2.3 UART/USB převodník – UB232R.....	14
1.3 MOŽNÉ KONCEPCE	15
1.3.1 Původní koncepce	15
1.3.2 Prostředí Arduino	15
1.4 MIDI.....	15
1.4.1 Standard MIDI File.....	16
1.5 ŘÍZENÍ HARMONIKY	17
1.5.1 Velikost klaviatury	17
1.5.2 Tlak vzduchu	17
2. HARDWARE.....	18
2.1 KONCEPCE HARDWARU	18
2.2 RASPBERRY PI 3B+.....	18
2.3 ATMEGA 165A.....	19
3. SOFTWARE	20
3.1 SÉRIOVÁ KOMUNIKACE	20
3.2 ŘÍDÍCÍ PROGRAM	20
3.2.1 Použité knihovny.....	21
3.2.2 Flow programu	22
3.3 PROGRAM MIKROKONTROLERU.....	22
3.3.1 Funkce inicializace komunikace	22
3.3.2 Funkce inicializace výstupů	23
3.3.3 Funkce komunikace.....	23
3.3.4 Spouštění not.....	23
3.4 NÁVRH WEBOVÉ STRÁNKY	23
3.4.1 Flask server.....	23
3.5 KOMUNIKAČNÍ PROTOKOL	23
4. REALIZACE.....	24
4.1 KOMPLETACE HARDWARU	24
4.2 OŽIVENÍ DPS	25
4.3 OŽIVENÍ RPI	27

4.3.1	<i>Konfigurace operačního systému</i>	27
4.3.2	<i>Vzdálený přístup</i>	28
4.4	SPOUŠTĚNÍ APLIKACE.....	28
5.	ZÁVĚR	29
	LITERATURA	30
	SEZNAM SYMBOLŮ A ZKRATEK	32
	SEZNAM PŘÍLOH	33

SEZNAM OBRÁZKŮ

Obrázek 1.1 Schéma zapojení plošného spoje	13
Obrázek 1.2 Deska plošného spoje	14
Obrázek 1.3 Analogie not, jejich frekvencí a čísla v MIDI [19]	16
Obrázek 2.1 Blokové schéma řídicího systému	18
Obrázek 2.2 Pinout sběrnice GPIO [20]	19
Obrázek 3.1 Vývojový diagram řídicího programu	21
Obrázek 4.1 Osazená deska plošného spoje.....	24
Obrázek 4.2 Náhled hrající soustavy	25
Obrázek 4.3 Detail akčních členů	25
Obrázek 4.4 Nastavení pojistek mikrokontroleru	26
Obrázek 4.5 Raspberry Pi Imager	27

.

ÚVOD

Tato práce spojuje hudbu a moderní elektroniku. Automaticky hrající hudební nástroje neboli orchestriony, jsou známy více než 200 let, jejich vývoj začal u mechanicky poháněných klavírů, houslí a perkusních nástrojů. Nosiče hudby byly například děrované papíry, když vzduch hnaný měchem prošel štěrbinou v papíru, aktivovalo se táhlo, které stisklo klávesu, či udeřilo paličkou. Dalšími nosiči mohly být válce pobité kovovými klínky, které mechanicky aktivovaly jednotlivé tóny nástrojů. Rozvoj elektrifikace poté rozšířil samohrající nástroje, které se nejčastěji objevovaly například v restauracích nebo na recepcích hotelů. Některé dražší a propracovanější stroje byly dokonce osazeny několika nástroji, například varhany, různé dechové nástroje anebo třeba i kontrabas, které hráli dohromady, skoro jako živá kapela.

V této práci je popsán postup vývoje elektronicky řízené obdoby automatického hudebního nástroje, konkrétně klávesové foukací harmoniky. Práce se zabývá návrhem kompletního řešení. Při návrhu je zároveň přihlédnuto k možnosti budoucího připojení dalších hudebních nástrojů.

Cílem práce je, aby navržený systém byl funkční, jednoduše ovladatelný a převozitelný.

1. TEORETICKÝ ROZBOR

Tato kapitola popisuje původní koncepci systému, která bude upravena nebo předělána ve zbytku práce. Zároveň jsou v kapitole popsány koncepce, které můžeme zvážit pro zhotovení systému.

1.1 Počáteční stav

Původní koncepce tohoto systému byla navržena jako diplomová práce Ing. Iriny Trukhiny **Chyba! Záložka není definována.**[1]. Některé části byly navrženy vhodně, a proto budou použity i pro tuto práci. Tato kapitola se zabývá těmito prvky a jejich vlastnostmi. Koncepce původního systému je také zmíněna v kapitole 1.3.1.

1.1.1 Foukací harmonika

Foukací harmonika je hudební nástroj oblíbený v žánrech jako country a folk. Zvuk se ve foukací harmonice vytváří tak, že hráč fouká do štěrbin, které směřují vzduch na různě dlouhé plátky (jazýčky), které se následně rozvibrují a tím vytváří jednotlivé tóny. Pro automatizaci je však vhodnější použít foukací harmoniku s klapkami (také známou jako melodika). Zde opět zvuk vzniká vibracemi jazýčků, ale vzduch je distribuován zmáčknutím příslušných kláves. Tento nástroj kombinuje akordeon a foukací harmoniku a je jednoduchý pro začátečníky.

Pro tuto práci bude použita melodika Hohner STUDENT 32 [2]. Tato harmonika je osazena 32 klapkami, což odpovídá rozsahu 2,5 oktávy. Konkrétní rozsah je F – C₃, díky čemuž bude harmonika vhodná pro velkou škálu melodií různých žánrů.

1.1.2 Elektromagnetická táhla Saia-Burgess 195224-223

O stisknutí kláves harmoniky se budou starat elektromagnetická táhla Saia-Burgess 195224-223 [3]. Pro 32 kláves klaviatury jich bude potřeba 32, tedy jedno pro každou klávesu.

Tyto solenoidy mohou být při 24V napájení a teplotě prostředí 20 °C aktivovány bez časového omezení. Při určeném napájení má každý aktivovaný solenoid odběr 7 W.

1.1.3 Zdroj 24 V

Maximální očekávaný odběr výkonu P_{MAX} je definovaný následujícím vztahem.

$$P_{MAX} = 32 * 7 + P_{CPU} + P_{DPS} = 240 W \quad (1.1)$$

Kde P_{CPU} je maximální odběr mikrokontroleru, který se obvykle pohybuje v řádu jednotek W. P_{DPS} je pak maximální odběr zbytku součástek na desce plošného spoje, který také očekávám v řádu jednotek W. Z toho vyplývá, že 240W zdroj, který byl k tomuto systému poskytnut je na hranici maximálního výkonu. V samotném řešení bude nutné

zajistit, aby se zdroj nepřetížil. Toho dosáhneme například stanovením maximálního počtu solenoidů spuštěných najednou.

1.1.4 Regulátor tlaku

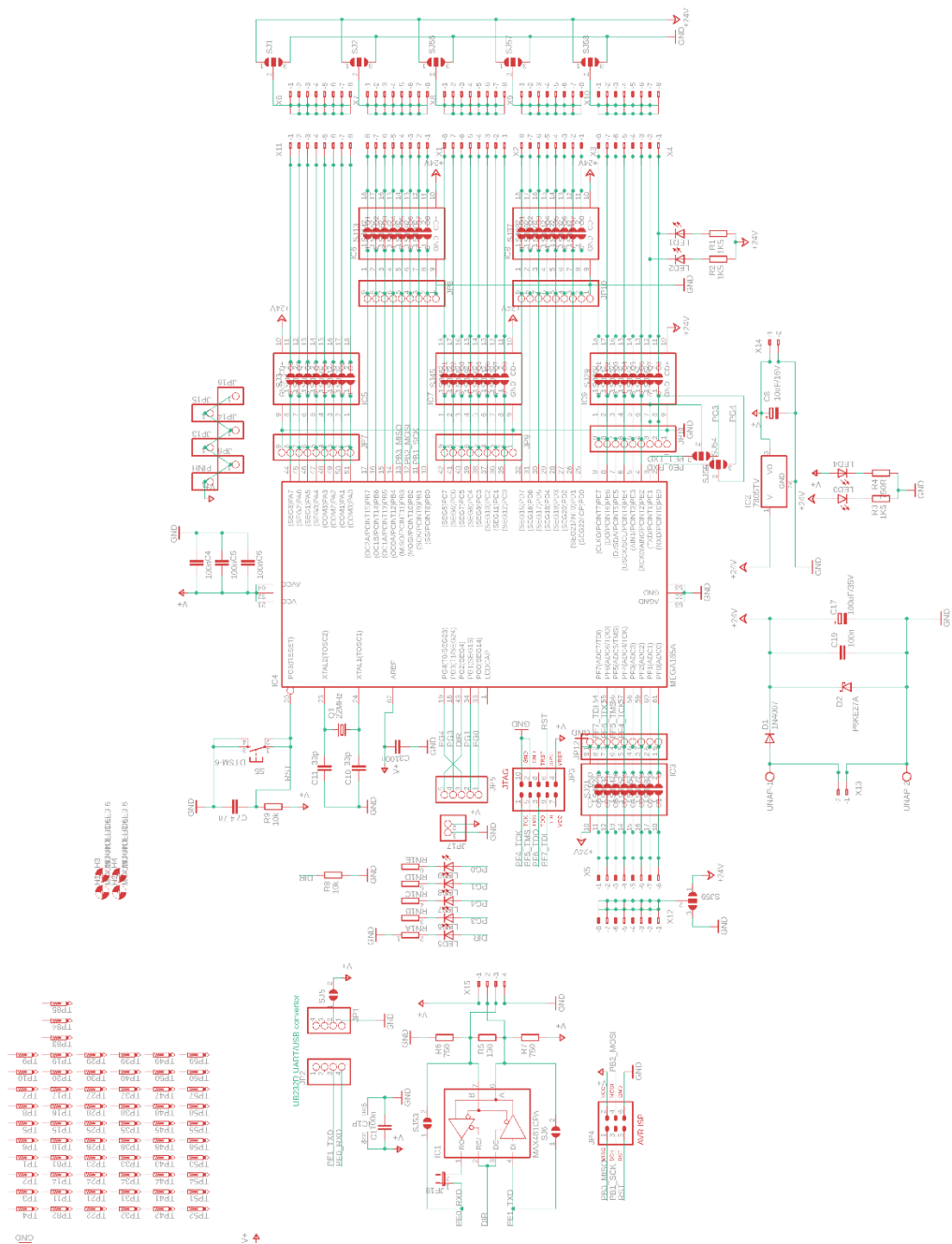
Z původní koncepce budeme také používat regulátor tlaku vzduchu ITV1030-31F1N [4]. Tento regulátor umožňuje regulovat tlak od 0,005-0,5 MPa. Je možné jej ovládat analogovým signálem 0–10 V. Regulátor umožňuje zpětnou kontrolu tlaku pomocí integrovaného měřiče tlaku. Tento výstupní signál je analogový v rozmezí 0–5 V.

1.2 Návrh DPS

DPS byl vytvořen s přihlédnutím k univerzálnosti desky, pokud by se koncepce ukázala jako nevhodná, nebo by bylo třeba připojit jiné periferie, než původní záměr projektu uvažuje.

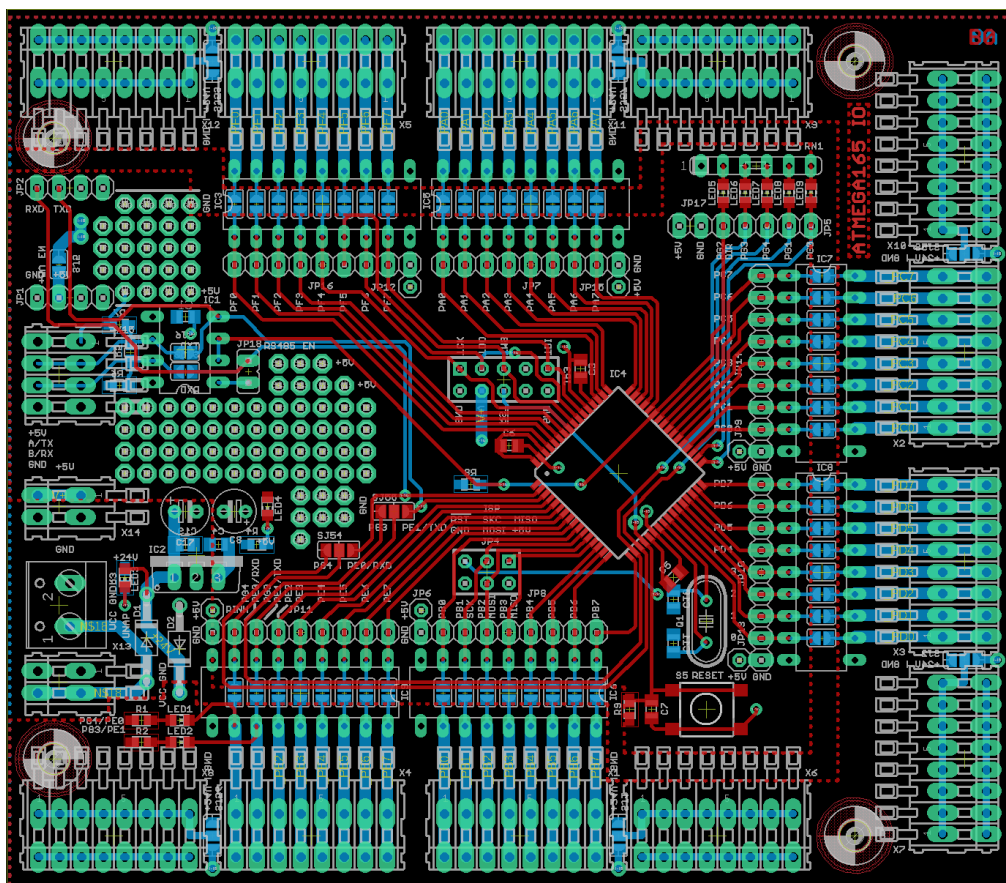
1.2.1 Schéma zapojení a deska plošného spoje

Zapojení desky plošného spoje bylo realizováno dle obrázku 1.1.



Obrázek 1.1 Schéma zapojení plošného spoje

Samotná deska plošného spoje je pak zobrazena na obrázku 1.2



Obrázek 1.2 Deska plošného spoje

1.2.2 Drivery ULN2803

Pro aktivaci táhel byly vybrány drivery ULN2803 [8]. Tato součástka je složena z 8 párů tranzistorů zapojených v takzvaném Darlingtonově zapojení se společným emitorem. Pouzdro součástky je DIP18. Tyto drivery jsou vhodné pro indukční zátěže a jejich parametry vyhovují požadavkům práce. Maximální odebíraný proud je 500 mA (táhla při 24 V odebírají necelých 300 mA), ovládací napětí je 5 V (optimalizováno pro TTL logiku) a ovládané napětí může být až 50 V. Pro ovládání 32 táhel potřebujeme 4 drivery.

1.2.3 UART/USB převodník – UB232R

Pro komunikaci mezi RPI a MCU bylo zvoleno rozhraní USB, to však MCU nativně nepodporuje. Proto byl použit převodník UB232R, který bude sloužit jako prostředník komunikace mezi deskami. Ze strany MCU bude tedy komunikace UART a ze strany RPI USB.

RPI sice umožňuje komunikaci UART přímo pomocí integrovaného pole vstupů a výstupů GPIO, avšak UART na RPI používá napětí 3.3 V a MCU používá 5 V. Z tohoto důvodu není možné tyto zařízení přímo propojit a je třeba převodníku. Alternativou tohoto řešení by mohl být DC/DC step-down modul, který by napětí 5 V snížil na 3,3 V

1.3 Možné koncepce

V této kapitole jsou popsány výhody a nevýhody jednotlivých koncepcí.

1.3.1 Původní koncepce

Koncepce z diplomové práce používala k řízení vstupů rozhraní GPIO, které je přímo integrované na desce Raspberry Pi. K ovládání akčních členů byly použity adresovatelné registry komunikující pomocí I2C sběrnice.

Výhodou této koncepce je programování pouze jednoho prostředí a to RPI (Python nebo C). Nevýhodou však byla neuniverzálnost řešení a vyšší cena z důvodu většího počtu součástek. Další nevýhodou bylo přetížení I2C sběrnice, což vedlo k omezené funkčnosti.

1.3.2 Prostředí Arduino

Použití například desky Arduino Mega jako hlavního kontroléru a doplnit jej o potřebné výkonové výstupy je další z možných koncepcí.

Její výhodou je množství volně dostupných knihoven pro práci s MIDI soubory. Další výhodou je programování pouze jednoho prostředí. Jelikož Arduino je mnohem levnější než RPI, byla by cena nižší než u původní koncepce.

Nevýhodou je, že není jednoduchá možnost doplnit tuto koncepci o další hudební nástroj spolupracující s harmonikou. Možná nevýhoda by mohl být omezený výkon CPU ATmega 2560 (oproti RPI).

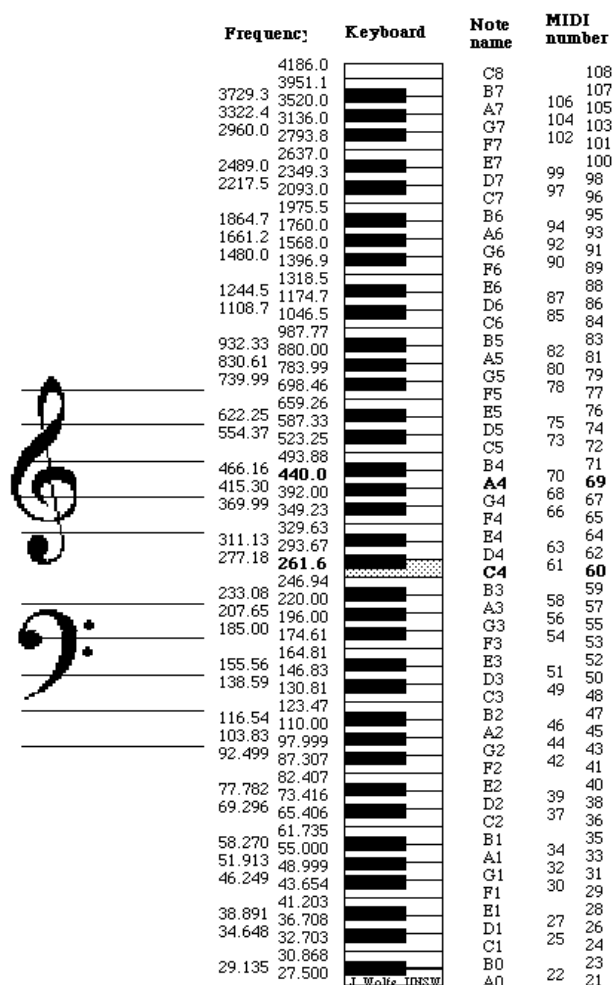
1.4 MIDI

Formát MIDI[10] je jednou ze stěžejních složek této práce. V této kapitole jsou popsány důležité vlastnosti MIDI.

MIDI neboli *Musical Instrument Digital Interface* vznikl v 80. letech minulého století v USA a jeho cílem bylo vytvořit volně dostupný standard pro komunikaci mezi hudebními nástroji různých výrobců. Do té doby totiž každý výrobce hudebních nástrojů používal svůj vlastní standard komunikace mezi nástroji, což ztěžovalo hudebníkům výběr nástrojů. Pro tento standard vzniklo sdružení výrobců MMA (MIDI manufactures association), které otevřeně rozhoduje o věcech týkajících se tohoto standardu.

Prostřednictvím MIDI se přenáší informace o stiscích kláves a změnách potenciometrů, či sliderů. Tyto stisky jsou interpretovány jako jednotlivé noty. Syntezátory zvuků mají neomezené možnosti interpretace těchto not. Díky tomu může hudebník použít jako ovladač elektronické klávesy, launcher, nebo i obyčejnou klávesnici k počítači.

MIDI může nést informace až o 16 stopách různých nástrojů. Jednotlivé akce jsou uloženy jako *zprávy*, které obsahují informace o stisknutí konkrétní noty, její hlasitosti, délky trvání, vibrata, pozice (levý, či pravý stereo kanál) a další. Tyto informace jsou přenositelné z původního nástroje (například z pianina) na kterýkoliv jiný nástroj právě díky elektronickým syntezátorům nebo počítačům. Hudebník může nahrát melodii na piano a v postprodukci skladby může stejná melodie znít jako hraná například na akordeon.



Obrázek 1.3 Analogie not, jejich frekvencí a čísla v MIDI [19]

1.4.1 Standard MIDI File

Se vznikem standardu MIDI vznikl mimo jiné také standardizovaný formát souborů, který by vhodně ukládal jednotlivé zprávy a umožnil jednoduché nahrávání a práci s MIDI formátem. Standardní formát souborů pro MIDI má příponu .mid. Tyto soubory jsou velmi úsporné, což pomohlo jejich velkému rozšíření. Písně jsou uloženy jako sekvence jednotlivých zpráv. Obsahují zároveň informace o časování jednotlivých akcí a jsou díky tomu jednoduše čitelné i na méně výkonném hardwaru. Soubory se skládají z hlavičky a těla.

V hlavičce souboru jsou obsaženy informace jako název skladby, počet stop (jednotlivých nástrojů), tempo, a typu uložené písně. Tyto typy písní jsou následující:

1. Typ 0 – všechny zprávy jsou uloženy v jedné stopě
2. Typ 1 – všechny stopy začínají ve stejný okamžik
3. Typ 2 – každá stopa je nezávislá na ostatních (málo používaný)

V těle jsou uloženy jednotlivé stopy, které začínají nultým řádkem s informací o nástroji, poté následují jednotlivé zprávy note-on a note-off.

1.5 Řízení harmoniky

Pro správnou přípravu ovládacích programů bylo nutné stanovit několik omezení a předpokladů souvisejících s konkrétním objektem řízení.

1.5.1 Velikost klaviatury

MIDI soubory, které budou použity jako nosič melodií mohou obsahovat noty větší škály než koncertní piano ($A_0 - C_8$), konkrétně $C_{-1} - G_9$. Těmto notám jsou přiřazeny čísla od 0 do 127. Řízená harmonika má však klaviaturu pouze od $F - C_3$, tomu v MIDI odpovídá rozmezí od 53 do 84. Jiné noty jednoduše zařízení hrát nebude.

1.5.2 Tlak vzduchu

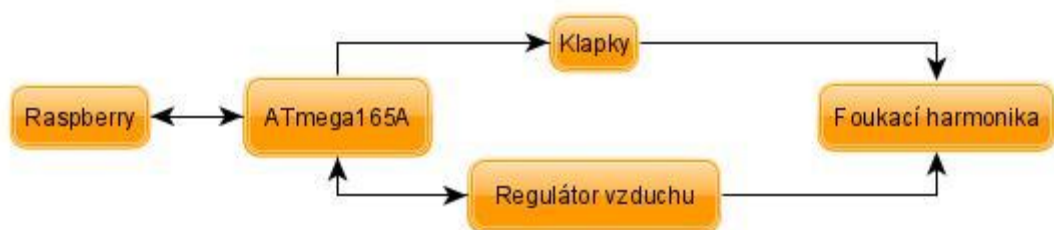
Foukací harmonika potřebuje k hraní zdroj vzduchu. Dechové hudební nástroje jsou dimenzovány pro tlak, který je schopný vyvinout člověk, bude tlak regulátoru omezen na rozmezí okolo 0,1 – 0,12 MPa.

2. HARDWARE

V této kapitole jsou popsány jednotlivé části hardwarového řešení, které nebyly popsány v dřívější kapitole.

2.1 Koncepce hardwaru

Koncepce, která byla pro realizaci vybrána je blokově naznačena na obrázku **Chyba! Nenašel jsem zdroj odkazů..**



Obrázek 2.1 Blokové schéma řídicího systému

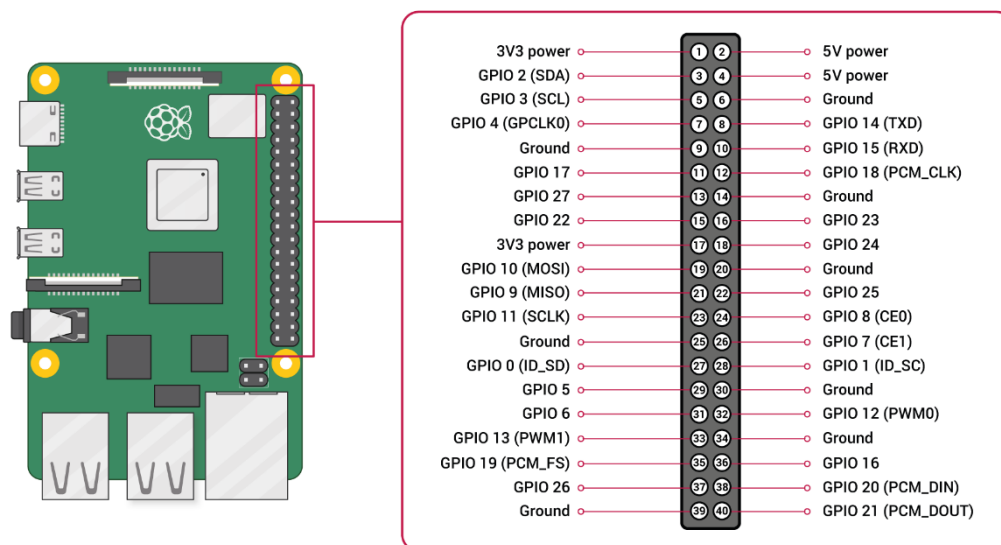
RPI komunikuje s mikrokontrolerem pomocí řetězce USB – UB232R – UART. Mikrokontroler přímo ovládá drivers, které spínají táhla klapek.

Tato koncepce je vhodná zejména díky svojí univerzálnosti (jednoduše se dá použít pro jiné hudební nástroje, stačí vyměnit akční členy).

Nevýhody této koncepce je nutnost seznámit se s programováním jak RPI, tak mikrokontroleru.

2.2 Raspberry Pi 3b+

Raspberry Pi 3b+ je třetí generace miniaturních počítačů, které se jsou vhodné pro širokou škálu DIY projektů [5]. Deska je osazená 64bitovým procesorem architektury ARM Cortex-A53. Dále deska disponuje například grafickou kartou (možnost připojení obrazovky), 1 GB RAM paměti, čtyřmi USB porty, Wifi, Bluetooth anebo nízko úrovněm rozhraním GPIO [6].



Obrázek 2.2 Pinout sběrnice GPIO [20]

RPI bylo zvoleno jako základ koncepce z důvodu možnosti rozšíření projektu o další nástroje. Dále je velká výhoda RPI výše zmíněná konektivita pomocí Wifi, díky níž bude možné vše řídit bezdrátově a bez nutnosti být fyzicky poblíž zařízení.

2.3 ATmega 165A

Mikrokontrolerem, který bude ovládat akční prvky projektu, byl zvolen 8bitový ARM mikrokontroler ATmega 165A od firmy Atmel. Tento MCU je pro tento projekt vhodný, protože má dostatečně vybavené rozhraní, pro všechny potřeby projektu [7].

Vlastnosti MCU, které mohou být využity jsou například PWM výstup (pro ovládání regulátoru tlaku), programovatelné I/O piny pro ovládání táhel, další výstupy pro indikační LED.

Frekvence krystalového oscilátoru byla zvolena 11,0592 MHz, protože je možné z této frekvence násobením a dělením odvodit většinu frekvencí běžně používaných UART rozhraním.

3. SOFTWARE

Software zařízení se skládá z dvou hlavních částí, jedna ovládající mikrokontroler ATmega a druhá, která běží na raspberry a zprostředkovává přehrávání MIDI a komunikaci s uživatelem.

3.1 Sériová komunikace

Mezi RPI a MCU bude probíhat sériová komunikace UART s touto konfigurací:

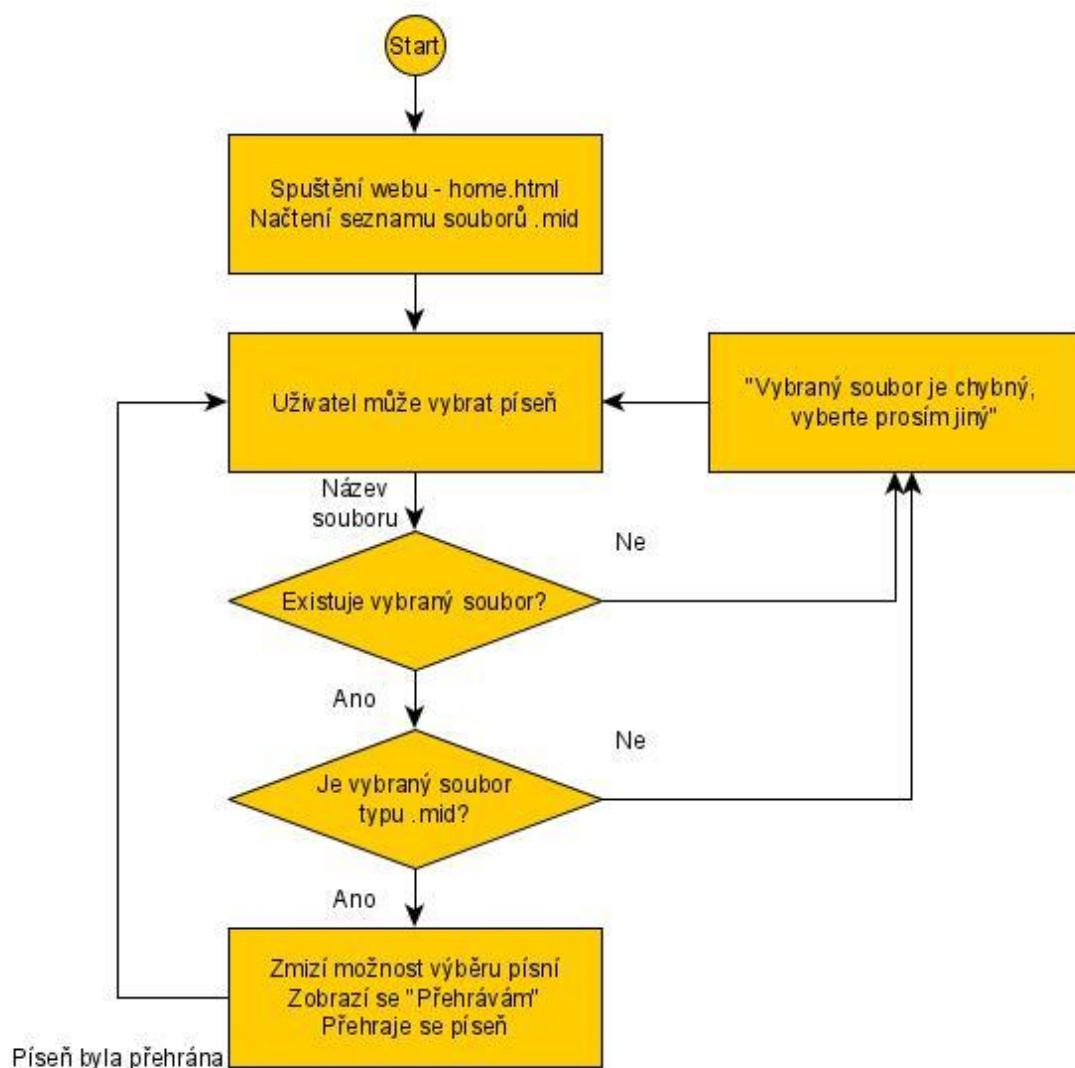
- Rychlost: 9600 baudů (bitů za sekundu)
- Počet STOP bitů: 1
- Parita: bez parity
- Velikost zprávy: 8 bitů

Dohromady se po sériové komunikaci budou posílat zprávy o délce 10 bitů (1 START bit, 8 bitů s daty a 1 STOP bit) s rychlostí 960 B za vteřinu. Tato rychlost je pro naše použití více než dostatečná. Rychlost 9600 baudů nebyla zvolena náhodou. Krystalický oscilátor, který jako zdroj strojového času pro mikrokontroler byl zvolen s frekvencí 11 MHz (11059200 Hz). Tato frekvence ve spojení s rychlostí 9600 baudů přináší 0% chybu časování.

3.2 Řídící program

Hlavní program, který bude číst a přehrávat MIDI soubory a komunikovat s uživatelem bude operovat na RPI. Tento program bude v jazyce Python[18] (verze 3 a novější). Společně s jazykem Python bude pro webovou stránku použit značkový jazyk HTML a Javascript.

Funkce tohoto programu bude načíst soubory v určené složce, určit, které z nich jsou MIDI a ty nabídnout uživateli. Poté co uživatel vybere, kterou píseň chce přehrát, program píseň načte a začne předávat povely mikrokontroleru o spuštění jednotlivých not.



Obrázek 3.1 Vývojový diagram řídicího programu

3.2.1 Použité knihovny

Jazyk Python byl zvolen právě kvůli dostupnosti mnoha knihoven přímo upravených pro Raspberry Pi a mimo jiné i pro práci s MIDI soubory.

Všechny knihovny budou nainstalovány pomocí balíčkového správce *pip* (package installer for Python).

Z knihovny *Mido*[11] bude použit objekt *MIDI Files*, který umožňuje načíst MIDI soubor a pracovat s ním jako s objektem. Mezi funkce tohoto objektu patří například:

- *Save()* – uloží objekt jako soubor MIDI
- *Play()* – vypíše jednotlivé zprávy postupně se zpožděním, které je shodné s časováním jednotlivých not v melodii.

Práce s MIDI objekty pomocí *MIDI Files* umožňuje také například výpis délky stopy, její typ, počet stop nebo tempo písně. Tato knihovna byla ale zvolena právě pro funkci *play()*. Tuto funkci využijeme právě pro vypsání MIDI zpráv přímo se správným

časováním jednotlivých not a ušetříme desítky řádků kódu. Funkce *play()* vypisuje každou zprávu jako n-tici prvků ve tvaru: note on/off, channel: 0–15, note: 0–127, velocity: 0–127. Touto n-ticí může listovat pomocí *tuple*.

Další knihovna, kterou budeme používat je knihovna *serial*[12]. Tato knihovna bude vhodná ke zprostředkování sériové komunikace mezi RPI a MCU. Z této knihovny budou použity funkce pro otevření portu, odeslání zprávy a přijetí zprávy. Protože jsou drivery pro převodník UB232R nainstalovány automaticky, není třeba žádné další nastavování. Stačí pomocí programu *serial.tools.miniterm*, který je součástí instalace knihovny *serial*, vyčist, který z USB portů je právě aktivní a tento port otevřít.

3.2.2 Flow programu

Po spuštění programu se načte spustí lokální web, kde se zobrazí stránka *home.html*. Na této stránce se zobrazí seznam spustitelných souborů s písněmi, které nejprve program zkontroluje, jestli jsou typu MIDI. Poté může uživatel zvolit píseň, kterou chce přehrát.

Po zvolení písně předá program název zvoleného souboru druhé části programu (*komAMidi.py*), která znovu zkontroluje, jestli je předaný soubor typu MIDI a jestli vůbec v příslušné složce projektu tento soubor existuje.

Pokud jedna z podmínek není pravda, hlavní program upozorní uživatele, že zadaný název souboru neexistuje. K této situaci by při správném používání však nastat nemělo.

Pokud je předaný soubor v pořádku, program jej načte a přehraje. Při přehrávání je uživatel upozorněn, že se právě přehrává píseň a není mu umožněno vybírat píseň jinou.

Po skončení přehrávání písně je uživatel vrácen na domovskou stránku, kde si může vybrat, kterou píseň chce přehrát jako další.

3.3 Program mikrokontroleru

Mikrokontroler ATmega 165A byl naprogramován pomocí programu Microchip studio[9]. Tento program umožňuje rozsáhlý debugging a vývoj programů pro AVR a SAM mikrokontrolery. Podporuje jak jazyky C/C++ tak strojový kód. Pro náš program byl zvolen jazyk C pro mikrokontrolery.

Program pro mikrokontroler bude mít mimo hlavní funkci *main* obsahující smyčku programu také 3 další funkce:

- *USART_Init* – nakonfiguruje sériovou komunikaci
- *USART_Transmit* – pošle byte do RPI
- *USART_Receive* – přijme byte z RPI

3.3.1 Funkce inicializace komunikace

Tato funkce nastaví jednotlivé registry mikrokontroleru dle výše zmíněné konfigurace sériové komunikace. Nejprve je nutné nastavit správnou rychlost komunikace do registru UBRR. Tuto rychlost můžeme zjistit z dokumentace MCU nebo pomocí vztahu

$$Ubrrr = \frac{F_{CPU}}{16 * Baud} - 1 = \frac{11059200}{16 * 9600} - 1 = 71 \quad (3.1)$$

kde F_{CPU} je frekvence hodin MCU a Baud je přenosová rychlost sériové komunikace.

Dále funkce spustí piny Tx a Rx. Nakonec se do registru UCSR0C nastaví délka slova (8 bitů) a počet stop bitů (1). Režim parity zůstane dle defaultního nastavení bez parity.

3.3.2 Funkce inicializace výstupů

Tato funkce zinicilizuje všechny potřebné výstupy a nastaví jejich hodnoty na nulu. Zároveň tato funkce může být zavolána pro vynulování všech výstupů.

3.3.3 Funkce komunikace

Funkce pro odeslání a přijímání jsou velmi podobné. Obě čekají na volno registru UCSR0A a poté buď vyčtou nebo zapíšou z/do registru UDR0.

3.3.4 Spouštění not

Po přijmutí zprávy od RPI převede program zprávu dle struktury switch-case na správný příkaz. Bude se jednat buď o požadavek na odpověď, díky níž RPI pozná, že je MCU připraven přehrávat, číslo noty, která se má sepnout nebo rozepnout nebo požadavek na vynulování výstupů MCU.

3.4 Návrh webové stránky

V této kapitole bude popsán návrh a fungování webového uživatelského rozhraní.

3.4.1 Flask server

Flask[13] je webový framework v jazyce Python, používající šablonování Jinja2, který nevyžaduje použití konkrétních podpůrných nástrojů nebo knihoven. Tento framework je relativně jednoduchý na používání a nabízí příjemnou integraci právě v aplikacemi v jazyce Python.

3.5 Komunikační protokol

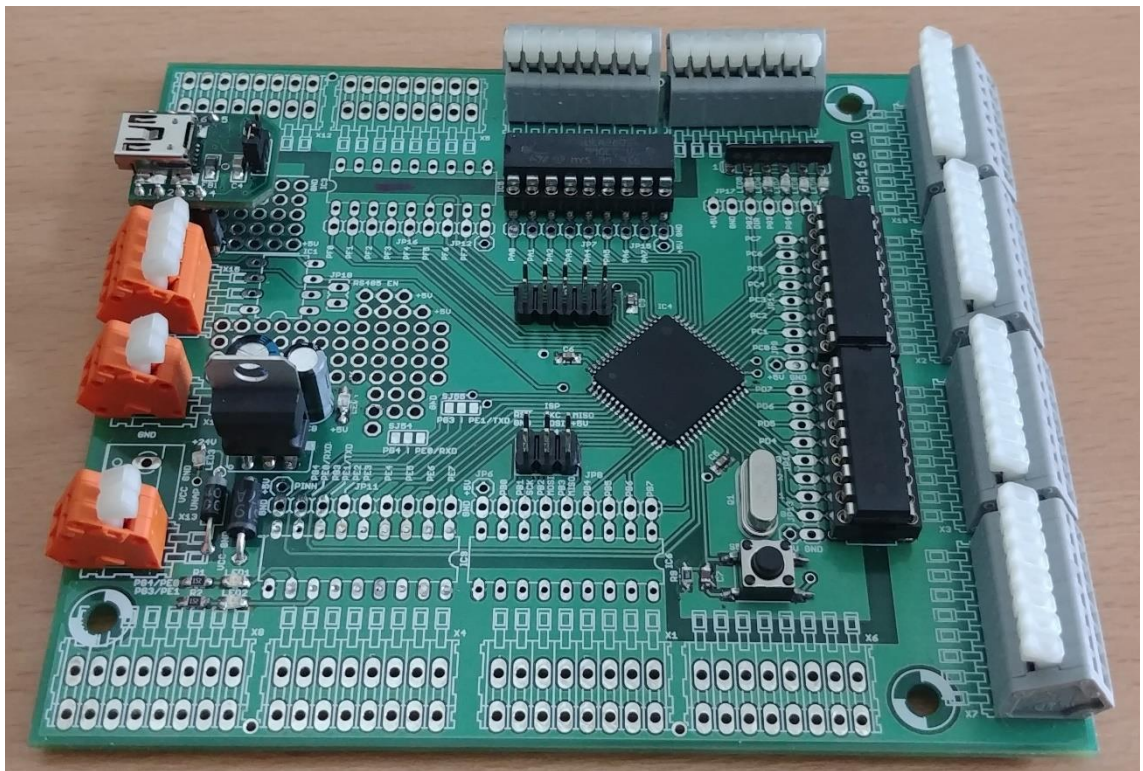
Pro komunikaci mezi RPI a mikrokontrolerem byl navržen komunikační protokol používající 8 bitové zprávy reprezentující znaky ASCII tabulky. Tyto zprávy budou přenášet čísla not, které se mají sepnout/rozepnout a několik znaků bude použito pro funkce řízení programu jako je třeba nulování výstupů nebo příkaz na ověření správnosti komunikace.

4. REALIZACE

V této kapitole jsou popsány jednotlivé kroky, které byly potřebné k finalizaci projektu. Dále jsou zde fotky hotového produktu. Bohužel zvuk ani video se na papír zachytit nedá. To je možná nápad na další práci.

4.1 Kompletace hardwaru

Jednotlivé komponenty, které byly pro tuto práci vyrobeny nebo byly převzaty a upraveny z původní diplomové práce jsou na následujících fotkách z průběhu vytváření práce.



Obrázek 4.1 Osazená deska plošného spoje



Obrázek 4.2 Náhled hrající soustavy



Obrázek 4.3 Detail akčních členů

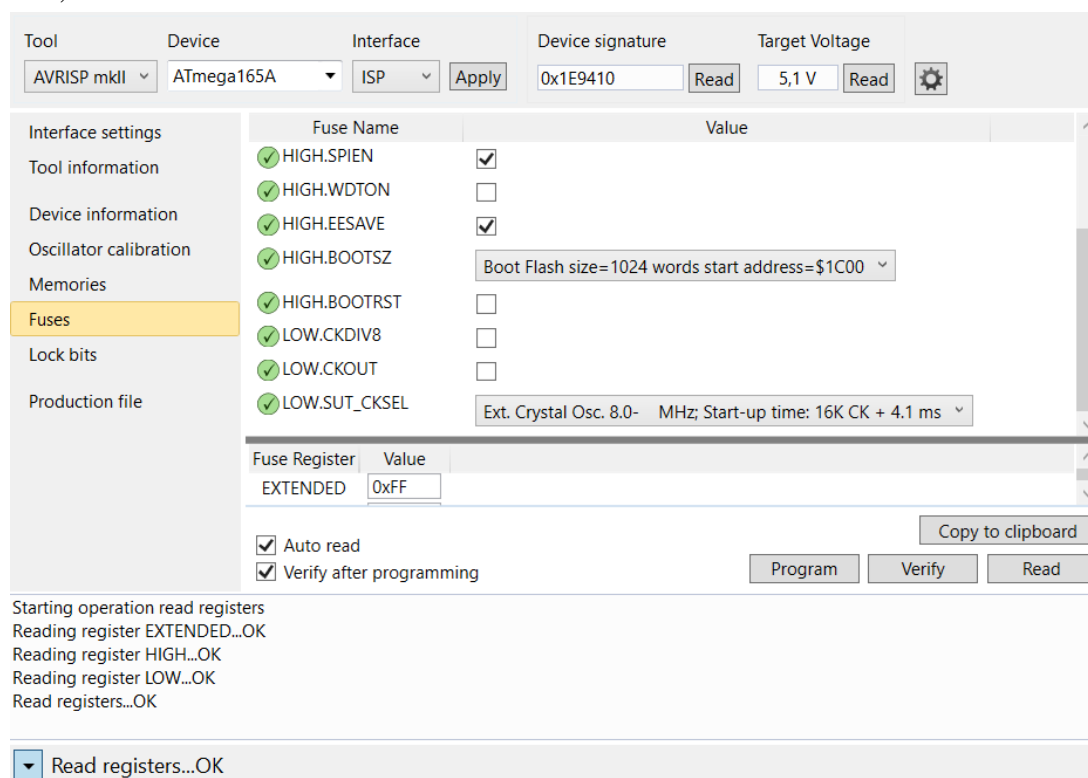
4.2 Oživení DPS

Po ověření, že jsou všechny potřebné součástky zapájeny na správném místě na desce bylo možné přistoupit k ožiování mikrokontroleru.

K tomu byl použit programátor AVRISP II[14], který byl připojen na připravené piny na desce tak, aby orientace konektoru byla shodná s orientací popisků na desce. Program Microchip Studio programátor rozpoznal ihned po připojení do USB počítače a provedl konfiguraci. Správnost spojení si můžeme ověřit v záložce *Device Programming* vyčtením napětí a čísla mikrokontroleru. Po ověření, že vše je správně připojeno jsme zůstali v záložce *Device Programming* kde jsme v záložce *fuses* vypnuli pojistku LOW.CKDIV8, která zapíná dělení strojového času osmi. Dále bylo nutné změnit pojistku LOW.SUT_CKSEL. U této pojistky se mění hned několik paramterů, první je rozmezí strojového signálu (8 MHz a více), další se týkají času, za který bude MCU předpokládat, že se krystalický oscilátor ustálil. Jelikož jsme neinvestovali do drahého a rychlého oscilátoru a nepotřebujeme extrémně rychlý start systému, zvolíme čas 16000 hodinových signálů + 4,1 ms. Ze vztahu

$$t_{START} = \frac{16000}{F_{CPU}} + 4,1 = \frac{16000}{11059200} + 4,1 = 5,55 \text{ ms} \quad (4.1)$$

kde F_{CPU} je frekvence hodin MCU a t_{START} je doba naběhnutí systému zjistíme, že i při zvoleném bezpečném čekání bude systém rozběhnout prakticky ihned. Na obrázku níže můžeme vidět záložku *fuses*, kterou je nutné zkontrolovat pro správné fungování zařízení.



Obrázek 4.4 Nastavení pojistek mikrokontroleru

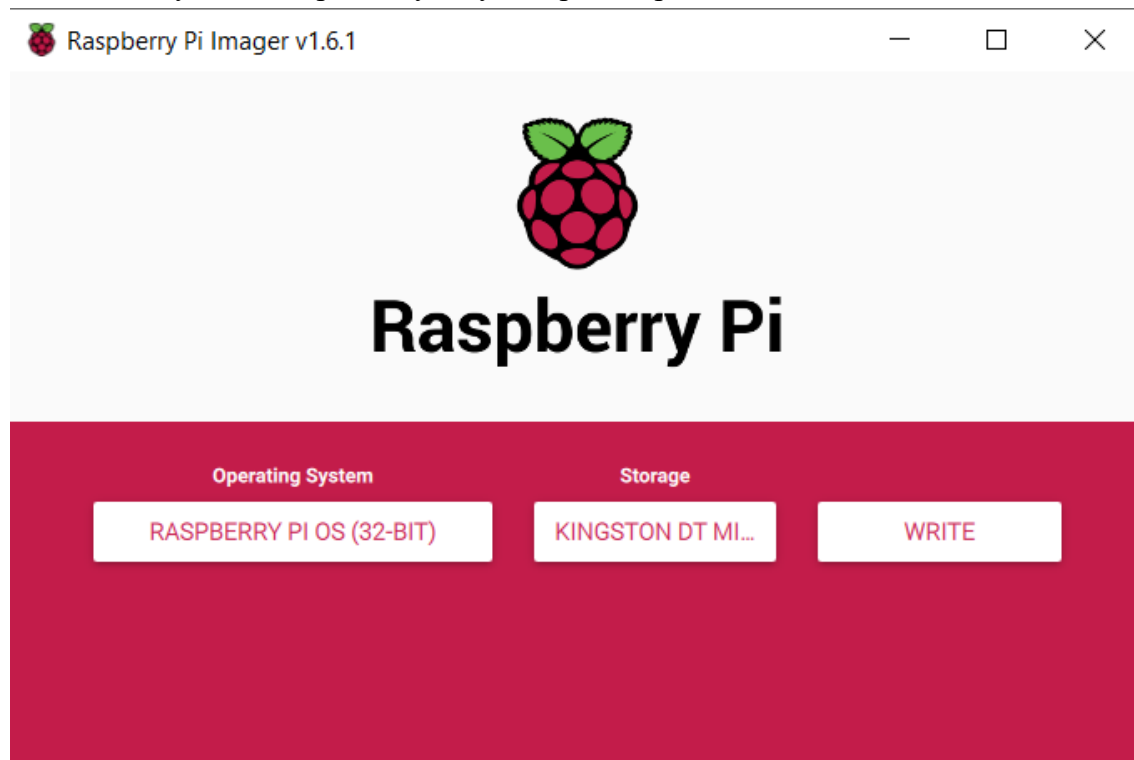
4.3 Oživení RPI

K programování RPI bylo nutné projít sérií úkonů, které jsou popsány níže.

4.3.1 Konfigurace operačního systému

Pro oživení raspberry byl použit program Raspberry Pi Imager, který je vyvinutý přímo pro vytváření bootovacích micro SD karet pro RPI. Program umožňuje připravení několika různých operačních softwarů vhodných pro RPI, většinou se jedná o ořezané verze desktopových linuxových distribucí (Ubuntu, Manjaro, ...) nebo o specifické distribuce, které byly vyvinuty pro konkrétní použití raspberry. Tyto speciální distribuce jsou například OctoPi (distribuce pro 3D tiskárny), RetroPi (distribuce pro emulaci starých her), Kodi OS (distribuce pro RPI jako multimediální stanici) a další.

Pro tento projekt byla zvolena distribuce Raspberry Pi OS[15]. Jedná se o distribuci založenou na jádru Debian OS s nástavbou Raspberry Pi Desktop. Tato distribuce je vyvinuta přímo pro použití v Raspberry Pi a má tedy už po prvním spuštění nainstalované téměř veškerý software potřebný k využití plného potenciálu RPI.



Obrázek 4.5 Raspberry Pi Imager

Po zvolení operačního systému je vhodné využít funkce pokročilého nastavení, která nám umožní nastavit hostname raspberry, povolení SSH protokolu, konfiguraci Wifi sítě anebo zvolení rozložení klávesnice. Do tohoto nastavení jsme se dostali stiskem kombinace kláves Ctrl + Shift + X. Díky nastavení výše zmíněných prvků nebylo nutné k Raspberry připojovat klávesnici, myš a monitor. Po vložení micro SD karty a připojení

napájecího zdroje proběhla instalace operačního systému a bylo možné se k RPI připojit pomocí SSH.

4.3.2 Vzdálený přístup

K programování RPI bylo nutné se k němu vzdáleně připojit. K tomu byl použit SSH protokol, který byl povolen při konfiguraci operačního systému. Pro toto připojení je možné použít libovolný SSH klient, například PuTTY, Dropbear nebo Bitvise. Pro tuto komunikaci je možné využít jak fyzické propojení pomocí ethernetového kabelu (hostname pro připojení je defaultně *raspberrypi.local*) nebo námi zvolenou Wifi. Pokud se chceme připojit k RPI pomocí Wifi, musíme nejdříve zjistit, jestli se RPI k síti vůbec připojilo (pokud ne, zkusíme opakovat konfiguraci operačního systému, nebo se připojíme přes Ethernet) a jestli ano, je nutné zjistit přiřazenou IP adresu. K té se dostaneme například přes nastavení Wifi routeru.

Po zjištění IP adresy jsme se pomocí programu PuTTY[16] připojili k RPI. Po přihlášení (defaultní přihlašovací údaje jsou *pi:raspberry*) jsme jako první změnili heslo k přihlášení do Raspberry. Poté jsme pomocí balíčkového nástroje APT nainstalovali program Xrdp[17]. Tento program nám umožňuje připojit se k RPI jako ke vzdálené ploše a používat klávesnici, myš a monitor našeho stolního počítače nebo laptopu jako ovládací prvky Raspbian OS. Xrdp používá mimo jiné právě protokol RDP(Microsoft Remote Desktop Protocol), který je předinstalovaný na všech počítačích s OS Windows. Přihlašovací údaje ke vzdálené ploše jsou stejné jako u SSH.

4.4 Spouštění aplikace

Ke spouštění aplikace bude vytvořen shellový skript, který se na RPI bude pouštět automaticky po zapnutí. S tím se zároveň spustí lokální server a pokud bude RPI připojeno na stejnou síť jako zařízení, ze kterého chce uživatel automatickou harmoniku ovládat, stačí se pouze připojit na IP adresu RPI.

5. ZÁVĚR

V této práci je popsáno kompletní řešení automatického systému hrající na foukací klávesovou harmoniku. Tato práce navazuje na diplomovou práci Ing. Iriny Trukhiny a přebírá některé koncepty ze zmíněné práce. Konkrétně se jedná o akční členy, 32 elektromagnetických táhel, která jsou použita k spínání kláves harmoniky a o konstrukci, která drží tyto akční členy ve správné pozici společně s harmonikou. Ve spodní části této konstrukce je uložena elektronika řídící celý systém společně se zdrojem energie. Jako zdroj vzduchu je nutné připojit externí kompresor, který bude poskytovat vzduch do regulátoru tlaku, ze kterého půjde vzduch do harmoniky.

Koncepce systému byla zvolena v podobě řídicího počítače (Raspberry Pi 3B+) komunikujícího s mikrokontrolerem ATmega 165A pomocí sériové komunikace. Mikrokontroler pomocí tranzistorových polí ULN2803 ovládá výše zmíněná elektromagnetická táhla. Práce také popisuje, jak mikrokontroler nastavit po prvním spuštění.

Raspberry Pi poté řídí celý běh systému. Poskytuje uživatelské rozhraní v podobě webové stránky, pomocí kterého může uživatel vybrat písničku, kterou chce od systému přehrát. Dále zajišťuje přečtení a přeložení MIDI souboru, který si uživatel zvolil pro přehrání a následně zajistí, že se soubor správně přehraje.

K této práci bude do budoucna žádoucí vytvořit webové stránky, které by nabízely více možností ovládání, například přidání, či odebrání písničky, nebo i přímo webový editor MIDI souborů. Zároveň by měli aktualizované webové stránky měli mít lepší grafickou úpravu, aby vyzdvihli působení celého systému jako celku.

Dále by bylo možné rozšířit řídicí program o další funkce, které by zamezovaly nestandardním situacím a nutnosti restartu systému. K těmto situacím se však dostaneme až pravidelným používáním výrobku.

Celý systém byl navrhován s myšlenkou budoucího připojení dalších nástrojů, které budou také automaticky ovládány a mohla by tak vzniknout komplexní automatická kapela. K tomuto připojení napomáhá například vysoký výkon řídicího počítače a modulární řídicí program, který lze jednoduše rozšířit o ovládání dalších periférií. Nástrojů, které by se dali k takovému systému přidat je celá řada.

LITERATURA

- [1] TRUKHNA I. Elektronický systém řízení hudebního nástroje. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 2014. 68s., 6s. příloh. Vedoucí diplomové práce doc. Ing. ZDENĚK BRADÁČ, Ph.D.
- [2] Hohner Student 32 Melodica. *Muziker.cz* [online]. [cit. 2021-01-03]. Dostupné z: <https://www.muziker.cz/hohner-student-32-melodica>
- [3] 75M-STA-EN: Pull Tubular Solenoids [online]. [cit. 2021-01-03]. Dostupné z: <https://www.tme.eu/Document/e6f2be922f2eef5ff85325c47e73bd7a/75M-STA-EN.pdf>
- [4] SMC ITV1030-31F1N [online]. [cit. 2021-01-03]. Dostupné z: <https://www.smcpneumatics.com/ITV1030-31F1N.html>
- [5] Raspberry Pi – Wikipedie. *Wikipedia.org* [online]. [cit. 2021-01-03]. Dostupné z: https://cs.wikipedia.org/wiki/Raspberry_Pi
- [6] Raspberry Pi 3 Model B+ 64-bit. *Rpishop.cz* [online]. [cit. 2021-01-03]. Dostupné z: <https://rpishop.cz/raspberry-pi-3b/896-raspberry-pi-3-model-b-plus-64-bit-1gb-ram-713179640259.html>
- [7] ATmega165/325/3250/645/6450. *Microchip.com* [online]. [cit. 2021-01-03]. Dostupné z: http://ww1.microchip.com/downloads/en/DeviceDoc/atmel-8285-8-bit-avr-microcontroller-atmega165a_pa_325a_pa_3250a_pa_645a_p_6450a_p_datasheet.pdf
- [8] ULN2803A STMicroelectronics. *Tme.eu* [online]. [cit. 2021-01-03]. Dostupné z: <https://www.tme.eu/cz/details/uln2803a/drivery-integrované-obvody/stmicroelectronics/>
- [9] Microchip Studio for AVR and SAM devices. *Microchip.com* [online]. [cit. 2021-01-04]. Dostupné z: <https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices>
- [10] Musical Instrument Digital Interface. *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-5-20]. Dostupné z: https://cs.wikipedia.org/wiki/Musical_Instrument_Digital_Interface
- [11] MIDI Files. *Read The Docs* [online]. [cit. 2021-5-20]. Dostupné z: https://mido.readthedocs.io/en/latest/midi_files.html
- [12] PySerial. *Read The Docs* [online]. [cit. 2021-5-10]. Dostupné z: https://mido.readthedocs.io/en/latest/midi_files.html
- [13] Flask Documentation. *Flask* [online]. [cit. 2021-3-12]. Dostupné z: <https://flask.palletsprojects.com/en/2.0.x/>
- [14] AVRISP mkII. *Microchip* [online]. [cit. 2021-5-20]. Dostupné z: <https://www.microchip.com/developmenttools/ProductDetails/PartNO/ATAVRISP2>
- [15] Raqspberry Pi OS. *Raspberrypi* [online]. [cit. 2021-1-4]. Dostupné z: <https://www.raspberrypi.org/software/>

- [16] Download Putty. *Raspberrypi* [online]. [cit. 2021-5-20]. Dostupné z: <https://putty.org/>
- [17] Xrdb by neutrinolabs. *Xrdp* [online]. [cit. 2021-5-20]. Dostupné z: <http://xrdp.org/>
- [18] PILGRIM, Mark. *Dive into Python 3*. [New York, NY]: Apress, [2009]. The expert's voice in open source. ISBN 1430224150.
- [19] <https://newt.phys.unsw.edu.au/jw/graphics/notesinvert.GIF>
- [20] <https://www.raspberrypi.org/documentation/usage/gpio/images/GPIO-Pinout-Diagram-2.png>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

FEKT	Fakulta elektrotechniky a komunikačních technologií
VUT	Vysoké učení technické v Brně
RPI	Raspberry Pi
CPU	Procesor (Central processing unit)
DIY	Do it yourself – Udělej si sám
ARM	Advanced RISC machine
MCU	Mikrokontroler
DPS	Deska plošného spoje
DC	Stejnoseměrný proud

Symboly:

U	napětí	(V)
I	proud	(A)
R	odpor	(Ω)
P	výkon	(W)
T	teplota	($^{\circ}\text{C}$)
p	tlak	(Pa)

SEZNAM PŘÍLOH

Všechny přílohy zmíněné v tabulce níže jsou nahrány na CD přiloženém k této práci.

IO.brd	Deska plošného spoje
IO.sch	Schéma zapojení
blokove_schema.jpg	Blokové schéma systému
AVR.zip	Program pro MCU
Vývojový diagram.jpg	Vývojový diagram
DPS.jpg	Obrázek osazené DPS
bokohled.jpg	Obrázek výrobku ze strany
nahled.jpg	Obrázek výrobku z vrchu
flaskProject.zip	Programu pro RPI